NPS 54-86-0005

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THE BASIS FOR TECHNOLOGICAL LITERACY:

EXPERIMENTS IN EXPLORATION

by

NORMAN LYONS

August 1986

Approved for public release; distribution unlimited.
Prepared for: Naval Personnel Research & Development Center, San Diego CA 92152

NAVAL POSTGRADUATE SCHOOL
Monterey, California

RADM. R. C. Austin
Superintendent

David A. Schrady
Provost

The research summarized herein was sponsored by National Communications Systems.

This report was prepared by:

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS 54-86-0005	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable) 54	7a NAME OF MONITORING ORGANIZATION Naval Personnel Research & Development Center
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code) San Diego, CA 92152
8a NAME OF FUNDING/SPONSORING ORGANIZATION NPRDC; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MIPR #HC1001-6-10118
8c ADDRESS (City, State, and ZIP Code) San Diego, CA 92152	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

The Basis for Technological Literacy: Experiments in Exploration

12 PERSONAL AUTHOR(S)
Norman R. Lyons

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
TechnReport	FROM July 85 TO Sept 86	April 86	89

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The objective of this research is to identify general problem solving skills that distinguish effective computer problem solvers from less effective ones.
Technological literacy is an area of inureasing concern to the Navy. As computer technology becomes pervasive, it is necessary to understand why some individuals adapt readil to machines while others do no. We need to know more about the skills that form the basi for technological mileracy.
Contrary to our expectations, mental imagery or advance organizers seemed to have little to do with success at the problem solving task. There seemed to be two principal determiners of success. The first was the way the suject chose to constrain the task. Th second was the level of general problem solgind skill that the subject brought to the tas

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Norman R. Lyons	22b TELEPHONE (Include Area Code) (408) 646-2666; 22c OFFICE SYMBOL 54Lb

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

THE BASIS FOR TECHNOLOGICAL LITERACY: EXPERIMENTS IN EXPLORATION

by

Norman Lyons

Administrative Sciences Department
Naval Postgraduate School
Monterey, California 93943
April 1986

Summary

Problem

Technological literacy is an area of increasing concern to the Navy. As computer technology becomes pervasive, it is necessary to understand why some individuals adapt readily to machines while others do not. We need to know more about the skills that form the basis for technological literacy.

Objective

The objective of this research is to identify general problem solving skills that distinguish effective computer problem solvers from less effective ones.

Approach

Subjects were given a picture drawing task using MacPaint and an Apple Macintosh. Subjects were given minimal instruction in the use of the system. To complete the task, they had to explore the system to learn to use the necessary features. A pretest and posttest questionnaire were used to determine the subjects' skills and backgrounds. The exploration sessions were videotaped using a split screen system that allowed observation of both subject and computer screen.

The subjects ranged from an eight year old girl to a sixty-eight year old man. Some subjects had experience with computers or word processors. None had any prior experience with a Macintosh.

Results and Conclusions

Contrary to our expectations, mental imagery or advance organizers seemed to have little to do with success at the problem solving task. There seemed to be two principal determiners of success. The first was the way the subject chose to constrain the task. The second was the level of general problem solving skill that the subject brought to the task

Future Research Considerations

A future study should break subjects into various groups of interest. Such groups would include grouping by age, sex, and prior background. Within these target groups, try to develop taxonomies of problem solving styles. These could include willingness to

explore, different approaches to exploration and so on. Once these have been developed, testing could begin on a variety of experimental treatments for teaching exploration or technological literacy skills.

CONTENTS

LIST OF FIGURES

Introduction

One puzzle that one notices in teaching beginning computer courses is the puzzle of technological literacy. Some students seem to grasp the concepts immediately. They have little trouble fitting things together and generalizing their discoveries. For other students, the whole thing is an exercise in frustration. They never seem to be able to coordinate their knowledge. Each problem is viewed as unrelated to anything else. The whole computer effort is seen as confusing and frustrating. These students are never able to get things together.

This is sometimes cited as evidence for the need to teach technological literacy. The world is becoming more dependent on technology, and what is needed is a technologically literate citizenry who can cope with this new world. But as a teacher, one sometimes notices surprising things about this technological literacy. Occasionally it happens that the struggling student is a technologically literate engineer while the student who breezes through is a music major who has never touched a computer in her life. There is clearly more to technological literacy than simple exposure to technology.

In the first research report on general problem solving, Lyons and Knott [1984], identified several areas as candidates for further research. One such topic is in the intersection between computer science, human factors and psychology. This is the study of exploration and exploratory environments. This literature is represented in the papers by Anzai and Simon [1979], Carroll, et. al. [1982, 1982a, 1983, 1983a, 1984, and 1984a], Darlington, et. al. [1983], Lewis, et. al. [1982], Malone [1981, 1981a, 1981b, 1981c, and 1982], and Schrager and Klahr [1983]. This literature looks at the way people learn about an environment without formal instruction. They must take their clues from the environment, form hypotheses and act on the basis of these hypotheses.

This study is of interest to cognitive psychologists because it provides a way to study problem solving behavior in a controllable environment. It is of potentially greater payoff to the computer industry. The man-machine interface is a make or break thing for a computer system. In microcomputers, some programs are popular partly because of their ease of use (LOTUS 1-2-3 is a good example). Other programs that may be equally powerful but unappealing to use become commercial failures.

This work with exploratory environments investigates how people solve problems in a system where there are few formal clues to proper behavior. This has always been a feature of computer systems. The manual never tells you everything there is to know about a system. It simply cannot. The number of possible combinations in a dialog make that impossible. It has always been true that some people are inherently better explorers of a computer environment than others. But until recently, there has been no research into why this might be so. Microcomputers have made this a more pressing question. Surprisingly, video games have had an influence as well. The video games show that exploration of complex environments is feasible and even a reasonable thing to do. There are few manuals on video games. The widespread use of micros even by non-technical individuals makes a better understanding of exploration necessary.

The work on exploration in the computer science references is interested primarily in the interface design question. Examples of this literature include the papers by Brown [1983], Good, Whiteside, Wixon and Jones [1984], Jacob [1983], Morland [1983], Norman [1983] and Stohr and White [1982]. But, there is a flip side to this question that is of interest in this research. How can we teach people to be better explorers? Is this a teachable skill? Why are some people better explorers than others? This is the same question (applied to a narrower domain) that motivated our earlier study. As computers become more pervasive, these are questions that need further research.

Relation of Problem Solving to Navy Needs

The issues grouped under the term technological literacy are becoming increasingly important to the Navy. Computer technology affects every aspect of the Navy from offices to weapons systems. Because of the increasing cost of manpower, the push is on to automate office operations. The government cannot afford to operate with expensive manual systems. Manual operations and paper based systems are giving way to office automation.

At the management level, there are computerized decision support systems to aid in budgeting and planning. More and more people are becoming users and supervisors of computer systems. In the tactical Navy computers are making their impact felt as well. C3 and intelligence systems rely increasingly on computers to function. Human reaction times are too slow for modern warfare. The tactical environment is becoming increasingly computerized.

It is unlikely then that any Naval career, be it officer or enlisted will be untouched by computers. The challenge over the next decade will be to integrate computer education and computer literacy into the training programs of the Navy. Without this, we will not meet the needs for technological literacy in Naval personnel.

To do this, however, we need to understand the components of technological literacy. This is not an area that can be studied using conventional structured testing methodologies. Measures of technological literacy are harder to construct than they are in conventional areas like reading. We are not interested so much in testing conventional skills as we are in testing interest and attitudes. We need to know the skills a person brings to a job and how they affect abilities to do job. These studies should help to design training programs to meet Navy needs.

Other Research in Exploration

The research related to exploration and problem solving is diverse and scattered. It may be found in Computer Science, Cognitive Psychology, Educational Psychology, and Engineering. There is no one discipline where this literature is concentrated. found. There is also no one body of scientists or academicians who are interested in looking into this.

Because it is an interdisciplinary study, the literature can be hard to track and difficult to understand. In doing the literature survey for this research, I have divided the literature into three areas. The first is research related to problems solving, the second is purely exploration oriented research, and the third is research related to the interface between computer science and psychology.

Research Related to Problem Solving

There is a small literature in educational research in the study of problem solving. It does not seem to be an area that has attracted widespread interest when compared with the literature on more conventional subjects like mathematics or reading. One article by Post and Brennan [1976] dealt with an experiment in teaching geometry to two groups of tenth grade students.

The authors developed a test of problem solving ability and administered it as both a pretest and a posttest to two groups of tenth graders. One group had received instruction in the General Heuristic Problem Solving Procedure (presented in Figure 1) while the other did not. The two groups did not show any significant difference in problem solving ability. The authors offer a detailed discussion of the statistical methodology they used. This discussion would be appropriate for a beginning experimental methods course. Unfortunately, they tell us nothing about how the problem solving methodology was taught. Were students merely exposed to it? Did they get substantial practice with it? These are questions left unanswered. The conclusions in the article are too strong to be justified by the material presented.

Raaheim and Kaufmann [1974] tested 73 secondary school students using tests of the subjects' ability to manipulate their environment. In one test, the subjects were asked to tell how to save a kitten stranded at the top of a tree. The lowest branch on the tree was too high to reach. The tools that they had available were a strong drill, a

I. Recognition, Clarification and Understanding of the Problem.
1. Read the problem carefully.
2. Look up any words you do not understand.
3. What is the unknown? What are the data? What is the condition? What is given?
4. State the problem in your own words.
5. Break the problem into parts.
6. Draw a diagram to aid in clarification.
7. Accept or reject the specific problem tools as a problem for yourself.

II. Plan of Attack - Analysis
1. Gather data (facts, rules, relationships) which are necessary for solution.
2. Recall missing data, select relevant data from the problem statement, and generate new data if necessary.
3. Eliminate irrelevant data.
4. Decide on needed approach activities by noting obstacles to the solution of the problem.

III. Productive Phase
1. Find the connection between the data and the unknown. You may have to consider auxiliary problems if an immediate connection cannot be found. Do you know a related problem?
2. Generate a hypothesis or a number of alternative hypotheses (possible solutions of the problem).
3. Order your data in preparation for hypothesis testing.
4. Reject initial hypotheses that do not satisfy the conditions of the problem.
5. Select a remaining hypothesis for testing.
6. Construct an algorithm or develop a heuristic for the manipulation of data as an instrument for possible verification of a hypothesis.

IV. Validating Phase - Checking - Proving
1. Accept or reject the hypothesis by verifying or not verifying that it meets the conditions of the task.
2. Look back. Can you check the result? Can you check the argument? Can you derive the result differently? Can you use the result or the method for some other problem?
3. If you have rejected your hypothesis, select a remaining one for testing.

Figure 1: General Heuristic Problem Solving Procedure - Post

hammer, a pair of pliers and a plane. The "best" solution involved drilling holes in the tree trunk and then using the hammer, pliers and even the drill as "steps" to climb the tree. There were other tests in a similar vein. They seemed to test the subjects' ability to overcome functional fixity in solving problems.

Success on their tests correlated highly with IQ. Girls were significantly poorer problem solvers than boys. The authors attributed this to their unfamiliarity with the tools posed in the problems. A possible alternate explanation might be that girls are more strongly socialized to be "good". This would tend to cause them to ignore the rule breaking required in these problems. The authors conclude that general problem solving ability exists and is highly correlated with conventional measures of intelligence.

The paper by Reif and Heller [1982] is a prescriptive piece. They lay out a teaching methodology for physics based on the general problem findings in cognitive psychology. They outline a strategy for structuring knowledge and acquiring knowledge in physics. They present a detailed program for physics instruction. There is no empirical validation, but they do try to tie the points stated in their recommendations into previous empirical work.

A paper by Baird [1983] offers an interesting summary of the problem solving research in education. Baird's report is similar to ours in that he is surveying the literature to see what has been done in problem solving and what benefits may result from teaching problem solving. He concludes that the benefits from teaching generic problem solving are difficult to prove. He believes that general problem solving techniques are probably most effective when they are integrated into domain specific courses.

A last paper is that by Ruth Pitt [1983]. It is a comprehensive literature survey of the psychological literature on the development of general problem solving schema. The work concludes that the development of a GPS schema takes place between adolescence and early adulthood. She studied two groups of students, one group of tenth graders and one group of college students. She found that the tenth graders had limited ability to define problems and generate hypotheses. They did not seem to have a rich set of problem representations or methods for dealing with problems. The college students were much better. Their abilities approached that of professors on most measures. The findings here are preliminary, but there are some interesting implications. Pitt suggests that the time to teach general problem solving techniques is in early adolescence when such concepts are being formed. More mature students are likely to respond better to domain specific techniques. For better or worse, their skills may already be formed.

If these papers are indicative of the general level of interest in education in teaching general problem solving, then the level of interest cannot be very high. It would seem that the primary influence of the general problem solving work has been to stimulate new approaches to the teaching of conventional disciplines as in the Reif and Heller article. There is little work involving the teaching or study of general problem solving ability. It may be that the concept is too hard to define and that it is safer to stay with conventional disciplines. It may also be that the best approach to general problem solving techniques is to integrate them in conventional areas to improve retention and comprehension.

Exploration Research

An interesting literature studying problem solving is growing in the intersection between computer science, human factors and psychology. This is the study of exploration and exploratory environments. This literature is represented in the papers by Anzai and Simon [1979], Carroll, et. al. [1982, 1982a, 1983, 1983a, 1984, and 1984a], Darlington, et. al. [1983], Lewis, et. al. [1982], Malone [1981, 1981a, 1981b, 1981c, and 1982), and Schrager and Klahr [1983]. This literature looks at the way people learn about an environment without formal instruction. They must take their clues from the environment, form hypotheses and act on the basis of these hypotheses.

This study is of interest to cognitive psychologists because it provides a way to study human problem solving behavior in a controllable environment. It is of potentially greater payoff to the computer industry. The quality of the man-machine interface can be crucial to the success of a piece of software. In microcomputing, some programs are popular partly because of their ease of use (LOTUS 1-2-3 is a good example). Other programs that may be equally powerful but unappealing to use become commercial failures.

Carroll [1982a] compares learning to use a word processor with the fantasy game "Adventure". Both are computer programs. The Adventure program is intentionally confusing, but it draws users into the world it creates. In Carroll's comparison, many word processors offer the same kind of confusion as Adventure without offering a balancing incentive to push past the confusion and learn the system.

The problems noted by Carroll can be formidable obstacles in learning to use a computer system. Some people regard them as so intimidating that they are never able to master the machine. Others regard them as a challenge and even become compulsive about responding to it. Carroll offers a series of suggestions for system design to reduce the problems noted in Figure 1. These suggestions are given in Figure 2. Carroll regards his work as preliminary. These results were published in the *IEEE Computer*, and seemed generally interested in stimulating thought on software design issues.

Malone [1981] is studying the same kind of issues as Carroll. His work also focuses on the relationship between computer games and more conventional computer programs. He is interested in what makes games so intriguing and whether these features of the game interface could be transferred to other programs. He ran subjects through a series of learning game experiments and tried to abstract features of the game that worked best. From this, he developed a set of heuristics for designing enjoyable interfaces. These are given in Figure 3.

Responsiveness	When you do something, you get some feedback (at least informational).
Benchmarks	You can tell where you are within a given episode or session. You have the means for assessing achievement and development of skill.
Acceptable uncertainty	Being less than fully confident of your understanding and expertise is OK.
Safe Conduct	You cannot do anything too wrong.
Learning by doing	You do so that you can learn to do; you design a plan; you do not merely follow a recipe.
Opportunity	Most of the things you learn to do work everywhere. You can reason out how to do many other things.
Taking charge	If progress stagnates, something new is suggested or happens spontaneously.
Control	You are in control or at least have the illusion of being in control.

Figure 2: Exploratory Environment Characteristics - Carroll[1982a]

This work with exploratory environments is trying to find out how people solve problems in a system where there are few formal clues to proper behavior. This has always been a feature of computer systems. The manual never tells you everything there is to know about a system. It simply cannot. The combinatorics render that impossible. It has always been true that some people are inherently better explorers of this computer environment than others. But until recently, there has been no research into why this might be so. Microcomputers and video games have made this a more pressing question. The video games show that exploration of complex environments is feasible and even a reasonable thing to do. There are few manuals to video games. The widespread use of micros even by non-technical individuals makes a better understanding of exploration necessary.

The work on exploration in the references is interested primarily in the interface design question. But, there is a flip side to this that is of interest in this research. How

I. Challenge

A. Goal. Is there a clear goal in the activity? Does the interface provide performance feedback about how close the user is to achieving his goal?

B. Uncertain outcome. Is the outcome of reaching the goal uncertain?

1. Does the activity have a variable difficulty level? For example, does the interface have successive layers of complexity?

2. Does the activity have multiple level goals? For example, does the interface include scorekeeping?

II. Fantasy

A. Does the interface embody emotionally appealing fantasies?

B. Does the interface embody metaphors with physical or other systems that the user already understands?

III. Curiosity

A. Does the activity provide an optimal level of informational complexity?

1. Does the interface use audio and visual effects: (a) as decoration, (b) to enhance fantasy and (c) as a representation system?

2. Does the interface use randomness in a way that adds variety without making tools unreliable?

3. Does the interface use humor appropriately?

B. Does the interface capitalize on the users' desire to have "well-formed" knowledge structures? Does it introduce new information when users see that their existing knowledge is (1) incomplete, (2) inconsistent or (3) unparsimonious?

Figure 3: Interface Design Heuristics - Malone [1981]

can we teach people to be better explorers? Is this a teachable skill? This is the same question (applied to a narrower domain) that motivates this whole report. As computers become more pervasive, this is a question that deserves research.

Psychology and Computer Programming

There is a large literature in computer science and psychology that relates to computer programming and human factors. Only a small fraction of this literature is relevant to our study. Much of this literature concerns the way in the way people interact with complicated systems. This interaction may be by experts who are well educated in the technology of this system involved. Frequently these studies revolve around way in which to structure these systems so that experts can use them more effectively. This is the general focus of the human factors literature as it revolves around such topics as the design of aircraft cockpits so that pilots can use them effectively in stressful situations.

The Computer Science related literature is similar in that it deals with the difficulties involved in using certain classes of programming languages. The normal focus in both classes of literature is either design of systems for specialists, or the selection of individuals for training as specialists.

Neither of these focuses was of interest for our study. We were more interested in the way in which novices or relative novices would react in an exploratory environment. We wanted to deal with an environment that was unfamiliar to the subject. This meant that we could make very few assumptions about the abilities of the subjects.

In addition, we had only a limited time in which to deal with the subjects. This meant that we could not give them a complicated set of instructions as would be typical in a computer programming class. Hence, there was no actual programming involved in our experiments. Because of this focus of interest of ours, a much narrower subset of the diverse literature was actual relevant to our study. Two areas were of primary interest, the first was mental imagery, and the second was the educational psychology literature dealing with the subject of teaching computer programming.

Mental imagery deals with the images that people use to represent information. We were interested in mental imagery, because we wanted to find out if there were a set of mental images that were useful in organizing an exploratory process. There is no hard evidence, but there is folklore in the computer area that certain types of people are better disposed in learning computing than others. One example of folklore is the statement sometimes heard is that musicians make good computer programmers.

If this is true, then perhaps music offers a set of compatible mental images that aid in the learning of computer topics. Mental imagery is discussed in a series of articles by Kosslyn [1981], and by Kosslyn, Pinker, Smith, Shwartz [1979]. The latter article presents a theory of mental imagery. It outlines the general research directions taken and provides an overview of the empirical foundations of mental imagery. This article presented the operating basis for the mental imagery concept that we are trying to analyze in this research.

The educational psychology literature was much richer. In a series of papers, R. E. Mayer [1975, 1976, 1981] deals with the problems of learning a computer language. In the 1975 paper, the study explored how novices learn to interact with a computer. The research goal was to develop instructional guidance to aid beginners in using the computer effectively.

Mayer looked at how a beginner acquires knowledge about the computer. He saw the user as bringing a mental model to the learning situation. The model is used as a vehicle for learning new material. Mayer is primarily concerned with the effect of different learning sets (cognitive structure, mental models, prerequisite experience...) on the acquisition of computer skills. He experimentally manipulates the activation of mental models used by novice computer users by providing different instructional sets.

Three experiments were run. In all experiments subjects were tested by one of two methods. The generation test required subject to write a program from a stimulus statement. In the interpretation test subjects were asked to describe what the program accomplished.

The hypothesis tested was whether two mental models of how a computer system operated influenced the programming knowledge acquired. One learning set was a diagram model which contained familiar objects and activity as an analog to the computer. The other learning set was a flow chart system of geometric symbols of the computer system. The two learning sets differed in the amount of related knowledge the models contained from the subjects' own experience. They used a simplified version of FORTRAN.

Subjects receiving the diagram model performed better than those who did not on the interpretation test. The flowchart subjects performed better on the generation test. He also observed that the model seemed to help low ability subjects' performance more than it did the high ability subjects.

Mayer concluded that learning technical material is easier when the learner is given a model of the overall task. The model should be stated in familiar terms. understand. New material may be assimilated and organized more easily than without a model. He terms these models as "advance organizers." Since the novice users in his experiment showed qualitative differences in performance depending on the type of model used, Mayer concludes that some advance organizers are better than others and that it is related to the degree of model meaningfulness to the subject.

Mayer reported further on this work in 1976 in "Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order." He defines meaningful learning as receiving information, having a cognitive structure available to hold information, and actively processing the material.

In this study, he predicts that the amount of information encoded when learning new technical information will increase when

1. The learner has a meaningful mental model of the what is to be learned.

2. The new material is actively processed during the learning phase.

He further predicts that when these conditions are met, there will be a greater transfer of knowledge to similar problem domains. Mayer designed two experiments to look at the information learned and how it is structured. The experimental conditions involved the presence or absence of meaningful learning sets before, during, and after learning a computer programming language.

Nonprogrammer subjects were taught a simplified version of FORTRAN. The subjects were tested in one of two ways:

1. Generation test questions requiring correct language statements in the correct order.

2. Interpretation test questions requiring correct interpretion of language statements.

Two experiments were run with much the same results as the earlier study.

In 1981, Mayer reviewed research concerned with the learning of technical information. He relates this to his interest in teaching nonprogrammers to interact with the computer. He asserts that learning is facilitated when a conceptual framework is used to connect the new information to existing knowledge. He bases much of his argument for using conceptual models on Newell and Simon's model of Human Information Processing and on Ausubel's concept of advance organizers.

He makes the distinction between rote and meaningful learning. Meaningful learning involves the connection between the existing knowledge in long-term memory and the recently acquired information in short-term memory. Material learned by rote simply exist in the memories as isolated units of information with little or no connection with the existing knowledge structures in long-term memory. Meaningful learning facilitates understanding which in turn allows one to apply what is learned across a spectrum of situations similar to the circumstances under which it was originally learned.

He emphasizes meaningful learning as the ability to transfer learned knowledge to new situations. Meaningful learning occurs under the following conditions:

1. **Reception** - the learner must attend to the information to be learned in order to receive the information in short-term memory.

2. **Availability** - the learner must have an existing knowledge structure in long-term memory to process the information received in short-term memory.

3. **Activation** - the learner must actively use the concepts from long-term memory during learning to establish the connection of the new information.

Much of his research addresses the question of whether the learning and transfer of technical information is increased by providing the learner with a familiar organization plan for the new material. He presents evidence from other fields to support his approach. The idea is to provide an introduction that contains only the general ideas to be presented in the new material.

From the evidence presented it was concluded that:

1. To assimilate new material the learner must have some type of conceptual framework available during learning.

2. Studies show an increase in the amount of material recalled aind learned when the learner is given a meaningful conceptual framework during learning. The conceptual framework acts as an advance organizer for new material to be incorporated into the learner's existing knowledge.

3. The advance organizers are most effective when:

    a) Learning unfamiliar material.

    b) Used with low-ability subjects.

    c) Testing transfer of knowledge to new situations.

4. Generating useful advance organizers requires:

    a) Logical relations between the ideas presented in the material.

    b) Logical connections between the new material and the model containing familiar concepts.

Mayer discusses the use of concrete models in learning computer programming. Concrete models have been used in other studies to increase technical comprehension. A concrete model is a representation which is an analogy for the concepts in the material to be learned. A concrete model is tangible representation. It should ensure that the learner has a familiar conceptual structure. He experiments with concrete models of computer programming as advance organizers for nonprogrammers to use before, during and after learning a new programming language. He reviews his studies with concrete models and concludes:

1. Concrete models enhance transfer performance but seems not to affect simple retention of new material.

2. Concrete models are useful when introduced before and during learning. They are not useful afterwards because the subjects have already encoded material in rote manner.

3. Concrete models enhance ability to recall conceptual ideas in the newly learned computer languages.

4. Concrete models improve low-ability subjects the most. High-ability subjects may already possess a good conceptual model.

A related survey paper is the paper by Moran [1981], which presents an overview of the field of user psychology. This is an attempt to develop a psychology of user behavior. Research focuses on the cognitive aspects of the user interface in specific tasks. He believes that the usefulness of a system ultimately depends on the quality of the user-computer interaction.

He lists the components that determine user behavior as:

1. User's goal - how one organizes actions to reach goal.

2. Task structure- the rules (boundaries) of the computer system, language, etc. This is often viewed as the "user interface."

3. User's knowledge- knowledge of the rules (task).

4. User's processing limits - which strategies are adopted are determined by the limits of ability to process the information. i.e. Short-term and long term memory, ways to relieve memory load.

He believes we can improve user behavior by:

1. Altering the interface.

2. Offering instruction to the user.

3. Changing the user goals.

4. Compensation for user limitations.

He states that the dimensions of user behavior must be evaluated by:

1. Functionality

2. Learning

3. Time

4. Errors

5. Quality

6. Robustness

7. Acceptability

We should describe population of tasks associated with particular system to properly incorporate the user in the total design of the system.

Moran believes that users vary along two primary dimensions:

1. Amount of knowledge - this usually correlates with experience. The challenges faced by novice and experts users are different. The novice user is continuously learning more about the system to solve a particular computing problem. The expert user interacts more procedurally to achieve a goal set by the computing task. The expert tends to be using a highly developed skill. He claims performance differences marked among novices using different interfaces but performance differences seem to diminish with experience.

2. The type of tasks engaged by user - Expert users are engaged in programming while novices are simply using applications programs. Programming in the purest sense is harder to learn and more cognitively complex skill.

Moran's field of interest is the user interface. This is any interaction between the computer and the user, any part of the computer system that the user comes into contact with. It can be the physical machine or the perceptual aspects of the system. An important factor of the interface is the extent to which the internal structures of the system are understood by the user, that is, the user's overall conceptual view obtained from the behavior of the computer system. This is termed the user's conceptual model of how the system operates. This conceptual model influences how the user will organize his actions to complete a particular computing task. This is similar to Mayer's advance organizers.

There have been several studies of the proper approaches to studying human-machine interaction. One such is Brooks [1980]. He looks at use of behavioral methodologies to describe programming language features and cites problems with current research. He feels that a major problem is usually found in the selection of subjects and selection of materials.

He talks about the use of open-ended questions as way to assess the cognitive structure a programmer may have in a computing task. Protocol analysis is mentioned as a form of the "open-ended question" measure. He recommends that the most fruitful approach in the current state of this research area be a "successful characterization of the programmer interaction ... based on model of the process or processes used by a programmer in interacting with a program. The development of the cognitive processes involved in programming is therefore, likely to be a prerequisite to progress on these methodological issues".

Another paper on programmer - machine interaction is by Sheil [1981]. He shows that many accounts constituting the basis of using particular programming techniques are based informal expert opinion in computer science or on experimental evidence which is methodologically flawed. He emphasizes importance of claims which describe a psychological basis for such assertions about programming methodology but also points out the lack of empirically testable hypotheses to back them up.

Sheil asserts that many behavioral studies of programming behavior are premature in their claims to be measure universal underlying processes in programming. Since computing behavior is a highly developed skill and that the level of skill is dependent on the extent of the programmer's knowledge base, he suggests that programming behavior first be characterized using more introspective techniques. This tends to support the use of protocol analysis as a first step in identifying the user's knowledge structure.

Experimental Setup

Subjects

A variety of volunteer subjects participated in this experiment. The group included both novice computer users and experienced programmers. The novice subjects were mostly volunteers from the Administrative Sciences Department staff. The experienced subjects were volunteers from the computer center at the Naval Postgraduate School. Three children were also included among the subjects.

Apparatus

The primary experimental apparatus was an Macintosh microcomputer and mouse interface. The microcomputer included a keyboard which was not necessary for this experiment. The subjects used the MacPaint program to reproduce an image on the screen of the Macintosh. The program provides a screen drawing window and several drawing tools. MacPaint represents tools by icons on the left side of the screen. An icon is a picture that represents the function that the MacPaint program can perform.

The user selects an icon by moving the mouse pointer to the icon and clicking the mouse button. The subject then moves the mouse pointer to the screen and proceeds to use the selected function. Subjects could then produce a picture by using the selected icons and the mouse to draw on the screen.

The experiment was conducted in a classroom containing a two-way mirror and audio visual and video equipment. Subjects were seated at a table in front of the Macintosh computer. The table had ample space to allow free movement of the mouse. Two television cameras were used to record protocols. One camera was to the side and the other was behind the subject. They did not interfere with the task. Judging from the reactions during filming, the subjects did not seem to be conscious of the equipment.

The cameras recorded a split screen image. One photographed the subject and the other photographed the computer screen. In this way, the experimenters could observe the images as they evolved. The two images combined on video tape giving a complete record of the subjects' experiences. In addition, the drawings produced were saved to a disc and were also printed on an Imagewriter printer.

Procedure

The subjects were given a picture of a house drawn using MacPaint (see Figure 4). The subjects were instructed to reproduce the house as closely as possible. The house was a simple drawing. The elements of the drawing were chosen to exercise as many of the options on the icons of the Macintosh as possible. The elements making up the house ranged from simple to complex.

Subjects were given a brief description of the computer system before they were turned loose. They were shown how to select icons and how to erase the screen for backtracking. The subjects were encouraged to talk freely about what they were doing during the experiment. Surprisingly, few people chose to. They seemed to become engrossed in the task and did not say much about what they were doing.

During the testing session the experimenter monitored the session from behind the subject near the video equipment. The subjects' performance was evaluated from the video tapes of the experimental session.

The experimental methodology used in these studies is a traditional cognitive psychology protocol analysis. The study of human-machine interaction is at too. early a level to design structured tests. Protocol analysis allows observation of problem solving behavior. We use this to assess the strategies users bring to a computer problem solving situation.

The experiment was not designed to assess differences in aptitude. Evidence shows that programming aptitude involves large knowledge bases and is hard to measure. Aptitude depends on the individual's experience [Sheil 1981; Moran, 1981]. We tried to avoid giving an individual an advantage because of background. MacPaint and the Macintosh were chosen because they were easy to use with no experience.

The Macintosh is still a sufficiently unusual system that none of our experienced subjects had used one. Further, none of the experience factors that our subjects had applied directly to the Macintosh. The attitudes and background of the subject brought to the task were of greater interest. Specific skills were unimportant.

Pretest and posttest questionnaires were used to assess the experience of the subjects (see Appendix B). Of special interest were the impact of advance organizers [Mayer, 1981]. A further interest was the cognitive structures or mental imagery subjects brought to the task. In planning the experiments it was felt that mental imagery or advance organizers might have a significant effect on performance. Some of the questions in the pretest were designed to elicit the mental imagery that the subject might use.

In computing, there is much folklore but little hard evidence about the factors that go to make a good computer user. The dimensions of technological literacy are still largely unexplored.

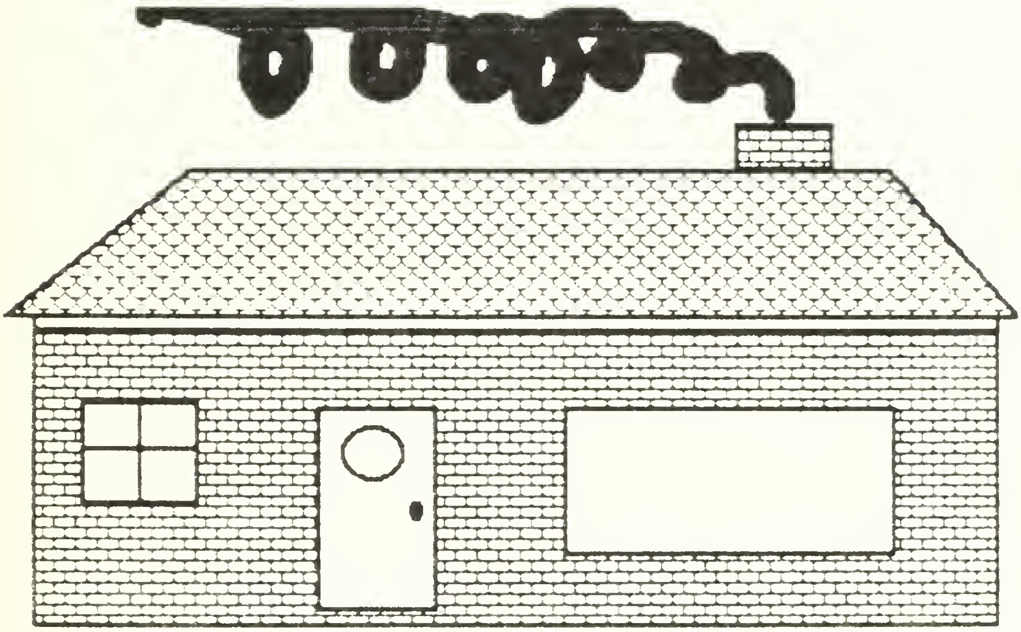

Figure 4: Macintosh House Drawing Problem

Before we can teach technological literacy, we need to know what it is. Quantitative measures will not help if we do not know what to measure. Protocol analysis is the best choice for experimental design here.

Discussion of the Experiments

The house drawing Figure 4 is a simple block drawing. The simplicity is deceiving, however. In Figure 5, we outline the house features and the tools the subject must use to draw the house features. The house features were selected to force the subject to explore the machine. It is not easily possible to select a single option, stay with it, and produce the house in Figure 4.

All subjects started with the walls of the house. This feature required texture and line thickness selection. To select texture, they must visit the menus at the side of the Macintosh screen. To select line thickness, the menu at the lower left corner is required. The subject selects texture using the menus at the bottom of the screen. The filled rectangle icon is required to draw the walls. This is selected from the menu at the side of the screen. Another approach is to use the blank rectangle and then use the paint pot icon for texture. None of our subjects use this approach, although one used the blank rectangle and tried to color it in with the pencil icon.

The ridgeline of the house requires the line thickness selection and a blank rectangle. If the subjects have selected a filled rectangle previously, they must reverse that selection. Must subjects either did not notice the ridgeline or deemed it an unnecessary detail. The small window required a line thickness selection, the blank rectangle icon, and the use of the rubber band line icon. Many of the subjects did not made transition from filled rectangle to blank rectangle. They ended up drawing a filled rectangle and then using the eraser icon to erase the filling.

The picture window required a line thickness selection and the use of the blank rectangle icon. The door required the same. The door window required a blank circle icon. This required the subjects to change shapes. The door knob required the filled circle icon. This meant the subjects had to change texture. The roof required a texture and line thickness selection. The subjects had to select a filled rubber band icon to draw the irregular shape of the roof. An alternate approach to this would be to select the paint pot and use blank rubber band shapes.

The chimney required texture and line thickness selections. This meant that using the filled rectangle icon. An alternative would be to use a blank rectangle and the paint pot icon. The smoke coming out of the chimney required the use of the pull-down menu.

The pull-down menu seemed to be the hardest feature for subjects to find. This is the set of selections at the top of the Macintosh screen. Normally, only the menu

House Feature	Macintosh Feature(s) Required
Walls	Texture and line thickness selection. Filled rectangle or paint pot and blank rectangle.
Ridgeline	Line thickness selection and blank rectangle.
Small window	Line thickness selection, blank rectangle and "rubber-band" lines.
Picture window	Line thickness selection and blank rectangle.
Door	Line thickness selection and blank rectangle.
Door window	Line thickness selection and blank circle.
Door knob	Filled circle.
Roof	Texture and line thickness selection. Filled "rubber-band" lines or paint pot and blank "rubber-band" lines.
Chimney	Texture and line thickness selection. Filled rectangle or paint pot and blank rectangle.
Smoke	Pull-down menu for brush thickness, texture selection and paintbrush.
Backtracking Commands	
Erase whole screen	Click twice on eraser icon.
Erase wide swaths	Select and use eraser.
Erase fine detail	Use pull-down edit window to select Fatbits option. Use move screen icon to move to details and pencil icon to draw or erase.

Figure 5: Macintosh Options Required

name is there. To activate them the subject must use the mouse to select an item and display its window. The subject must then systematically explore the pull-down windows until he finds one that changes the thickness of the paint brush. Most did not explore this thoroughly.

There is no obvious connection between items hidden in the pull-down menu and the selections that along the left side of the screen. In contrast many of the selections on the Macintosh, brush texture is difficult to find. In the pull-down menus, the Macintosh is more like a conventional computer, where options are hidden or only obvious to the expert.

The next important commands for this experiment are backtracking commands. In order to correct mistakes a subject must backtrack. The subjects demonstrated considerable willingness to use the backtracking commands. In the pretest, the subjects were shown how to do a couple of simple types of backtracking. There are three types of commands on Macpaint that allow the subjects to backtrack. The first enables the subject to erase the whole screen. This is done by clicking twice on the eraser icon.

This is a serious backtracking command. It destroys everything that the subject has done so far. The second backtracking command enables the subject to eraser wide swaths through the picture. This is done by selecting the eraser icon. When the subject selects that, a blank rectangle about a quarter of an inch on a side appears on the screen. The subject manipulates it by using the mouse. Every place that this blank rectangle touches is erased. If the subject has a steady hand, he is able to erase some detail, although this is difficult.

If the subject wishes to erase fine detail, then a more sophisticated method is required. The subject must use a pull-down window to select an option called Fatbits. This magnifies the screen and shows a pixel as a rectangle about two millimeters on a side. The subject then uses the pencil icon to draw or erase detail. The Fatbits option is difficult to find and no subjects used it intentionally.

Figures 6-a and 6-b give the scoring method used for judging the house diagram. The house diagram was judged on the basis of topological correctness. Topological correctness means that the subject's house should have the same shape as the original. It also had to have the same textures and the same pieces and the same relationship. It did not matter if the subject's house was the same size as the original. The criterion for judgement was shape, texture and the relationship of pieces.

The house was broken up into features. These were walls, ridgeline, small window, picture window, door, door window, door knob, roof, chimney and smoke. Each of these features was further broken down into the elements composing them. For example, the walls were broken down into shape, texture, and order. The shape meant that the wall had to be rectangular. Texture, in the case of the walls, meant that the subject had to select the brick texture. Border meant that the subject had to recognize that the

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	1
Texture	1
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1
Border	1
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	1
Texture	1
Roof	
Shape	1
Texture	1
Border	1

Figure 6-a: Scoring for House Drawing

Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	1
Texture	1
Total	29

Figure 6-b: Scoring for House Drawing

goal house diagram used a thicker line border than the computer was producing initially. The subject had to select a thicker line.

Each of these elements counted a score of one. To draw the house, the subject had to give his attention to each of these items in sequence. To do so the subject had to explore the computer and figure out how to shift the computer status from one item to the other. In the process of exploration, the subject had to be able to backtrack. In order to backtrack, the subject had to use the backtracking capabilities outlined in Figure 5.

For each subject, the house drawing was saved to a disk file. Later, the drawings were printed and used to compute individual scores based on the scoring method presented in Figures 6-a and 6-b. The scores for individual subjects may be found in the Appendix. There was one exception to this. One of the subjects was something of a perfectionist and was dissatisfied with the house produced. So, at the end of the experimental session, she clicked the erase button before the experimenter could save the house.

For this subject, we were able to score the house by reviewing the video recording. The performance of this subject was actually quite good. The subject was able to explore the computer, recover from errors, and backtrack effectively. It seemed that the subject's performance did not match up to her standards of excellence, and she was frustrated.

Figures 7-a and 7-b give the summary statistics for the subjects. The first part summarizes the results of the pretest. The detailed pretest and the instructions the subjects were given are in the Appendix. A total of 20 subjects were run. Twelve were male and eight were female. The age of the subjects averaged nearly 35 years with a standard deviation of 14. The oldest subject was 68 and the youngest was 7.

The average of number of years school was 14.8 with a range from 1 to 20. In this study, we were not trying for a homogeneous group of subjects. Since this was an initial exploratory study, we used a heterogeneous group. After we know more about exploration, we will be ready for homogeneous groups of subjects. Initial studies should use heterogeneous groups to isolate characteristics of interest.

Most of our subjects were familiar with typewriters. The average typing speed was a respectable 33.3. The range went from a maximum of 80 words per minute down to 0. The question about typing speed was included only to establish the subjects familiarity with computer-like equipment. Typing skill was not required in our experiment. All interaction is done via the Apple's mouse. The keyboard was available if the subjects wished to use it. For the purposes of the house drawing, there was no need. The video tape showed that most subjects pushed the keyboard out of the way during the experiment.

The next set of questions is our attempt to elicit information about advanced organizers. We asked subjects if they liked games, and 65 percent of them answered positively. The games reported in the answers (see Appendix) were varied. Subjects reported board games and card games most frequently. Surprisingly enough, video games came in at a much lower percentage. Only 40 percent of the subjects reported liking or playing video games. These reports came from the younger subjects. The older subjects were familiar with video games, but did not play them.

Fifty-five percent of the subjects reported themselves as comfortable with mathematics. One should be careful not to read too much into this. This is a self-reported level of feeling about mathematics and not an absolute one. One of the subjects who was uncomfortable with mathematics had a PH.D. in operations research from MIT. Presumably he is uncomfortable with mathematics at a higher level than most of the rest of us.

Only 45 percent of the subjects reported themselves as playing a musical instrument. None of the subjects reported any high degree of skill of music. Most of those who reported playing musical instruments were referring to skills learned in childhood and long since abandoned.

Seventy-five percent of the subjects reported that they knew how to use a typewriter. Since about half of the subjects came from the secretarial pool at the Naval Postgraduate School, this is not surprising. In addition, 75 percent of the subjects reported they knew how to use a word processor. For most people today, use of a word processor and use of a typewriter are synonymous. Typewriters are dying out.

Ninety percent of the subjects reported they had used a personal computer. Given the high level of penetration of personal computers in the schools and work place, this

Total Subjects:	20			
	Male	Female		
Sex:	12	8		
	Avg	STDV	Max	Min
Age:	34.8	14.3	68	7
Years of School:	14.8	5.3	20	1
Typing Speed:	33.3	24.5	80	0
House Score:	15.0	7.3	27	0

	Percent
Likes games:	65%
Video Games:	40%
Comfortable with Math:	55%
Musical Instrument:	45%
Use a Typewriter:	75%
Word Processor:	75%
Used a PC:	90%
Own a PC:	15%
Used a computer:	95%
Used a Macintosh:	0%

House Feature	
Walls	
Shape	80%
Texture	65%
Border	50%
Ridgeline	
Shape	40%
Texture	35%
Border	25%
Small window	
Shape	80%
Texture	90%
Border	45%
Crosspieces	70%

Figure 7-a: Summary of House Drawing Results

Picture window	
Shape	65%
Texture	75%
Border	35%
Door	
Shape	70%
Texture	85%
Border	40%
Door window	
Shape	55%
Texture	60%
Border	35%
Door knob	
Shape	10%
Texture	30%
Roof	
Shape	65%
Texture	55%
Border	35%
Chimney	
Shape	75%
Texture	55%
Border	40%
Smoke	
Pull-Down Menu Use	10%
Texture	20%

Figure 7-b: Summary of House Drawing Results

is not surprising. Only 15 percent of the subjects reported they own a personal computer. This is about twice the national average, and is about the level that marketing surveys expect to be the maximum penetration of personal computers in the marketplace.

Ninety-five percent of the subjects reported they had used a computer. This includes both personal computers and mainframe computers. This means that one of our subjects was a mainframe subject while all of the others had used either personal computers or a combination of mainframes and personal computers. None of our subjects had ever used a Macintosh before.

This is somewhat surprising in view of Apple's massive advertising campaigns since the introduction of the machine. Virtually all of our subjects were aware of the Macintosh, but apparently this awareness did not extend to trying to find an opportunity to use the machine. So, in this sense we started with a group of naive subjects. None of them had used this type of equipment before.

Since this was an exploratory study, we had no clear idea of what we might find as advance organizers. From the questions we asked in the pretest interview, no clear indicators of advance organizers emerged. The subjects liked games, but video games were not strong contenders. The games that they did mention as liking did not seem to be likely candidates for advance organizers in the task assigned. The use of musical instruments was low enough in the group that it does not seem to be a likely candidate either.

The most common area for people was prior use of computers and typewriters. Most subjects had experience with these machines, and during the test, they seemed to regard the Macintosh as just another example. The interviews and review of the video tape did not show any particular skill that the subjects were falling back on in order to do this task. Based on what we observed, the search for advance organizers in this task would have to be deemed a failure.

The subjects performance in house drawing, however, was anything but a failure. The average house score was 15 with a standard deviation of 7.3. The subjects scores range from a minimum of zero to a maximum of 27. Most subjects performed quite creditably on the task.

When we reviewed the performance on the individual subtasks, we see the subjects largely confined themselves to the major tasks involved in drawing the house. Most subjects were able produce the major features, the walls, the small window, picture window, the door, the roof, and the chimney. When we come to the less important features of the house, we find less compliance with the model. Only 40 percent of the subjects got the ridge line, 10 percent were able to reproduce the door knob, and only 10 percent were able to find the pull-down menu to create smoke correctly.

The subjects were shown how to use a pull-down menu in pretest orientation, but most of them either forgot or decided that this was not crucial to the test. In our reviews of video tapes, we have only a minority of the subjects using the pull-down menus. Even those who use them do not explore them fully. Most of the shapes involved in the house were rectilinear or polygonal. The subjects did well on these, and had no problems in shifting from one shape to another.

When the shapes were curved, as with the door knob and the door window, the subjects had more difficulty. In many cases, they missed the proper icons. In the case of the door knob, many subjects did not use the filled circle. They tried drawing the door knob free hand using a variety of other icons. When the results were not what they had hoped for, most of the subjects moved on, apparently considering the door knob too trivial to bother with. They already had substantial investment in drawing the door, and erasing the door knob would have destroyed that. This happened in virtually all of the subjects. Where the investment was viewed as substantial, there was no desire to backtrack.

Something of the same sort seems to have occurred in drawing the smoke. Apparently the smoke was seen as amorphous enough that there was no need to work hard in producing results like the model. This as well as the difficulty in finding pull-down menus may have been a reason for the low score for smoke. Subjects explored the pull-down menus but they did not try hard to duplicate the quality of smoke in the model house. Also, the smoke was the last task. They were nearing the end of the experiment and most were ready to move on.

In summary, the house task worked well in eliciting exploratory behavior from the subjects. In future experiments, I would recommend putting in more details to force more exploration. Details should be added that would force the subjects to transition from one icon to another in performing the task. This experiment did reasonably well at that.

Figure 8 presents Carroll's list of problems typically faced by subjects in an exploratory environment. It is interesting to compare this list with the kinds of problems faced by our subjects on the Macintosh. The Macintosh is a good test of exploratory environments. It is not as intimidating as other computers.

Carroll lists the characteristics disorientation, emptiness, and paradox. Disorientation means that the player does not know what to do in the environment. Emptiness means the computer offers few hints to help achieve goals. This was definitely not the case in the Macintosh. Disorientation was kept to a minimum by hints for action around the screen. Emptiness was not a Macintosh characteristic. Paradox was seldom a problem either. The Macintosh rarely told the player any course of action that was inappropriate.

Some of the other problems faced by subjects of exploratory environments were present. The illusiveness, mystery messages, slipperyness and side effects could also be a problem on the Macintosh. In observing the tapes of the subjects, we noticed that several of them managed to activate a menu feature called "Fatbits". There are two ways to do this. One way involves choosing that selection from a pull down menu. One subject did this on purpose. The subject explored the feature, decided it was not useful, and went on.

Disorientation	The user/player doesn't know what to do in the system environment.
Illusiveness	What the user/player wants to do is deflected toward other, perhaps undesired, goals.
Emptiness	The screen is effectively vacant of hints as to what to do or what went wrong.
Mystery Messages	The system provides feedback that is useless and/or misleading.
Slipperiness	Doing the "same thing" in different situations has unexpectedly different consequences.
Side effects	Taking an action has consequences that are unintended and invisible but cause trouble later.
Paradox	The system tells the learner/player to do something that is clearly inappropriate.
Laissez-faire	The system provides no support or guidance for overall goals (e.g. "winning" "typing a letter")

Figure 8: Typical Problems Faced by Users - Carroll[1982a]

The other way to activate it is to click twice on the pencil icon. It is easy to do this latter approach inadvertently. If the subject does this, the Macintosh zooms in on a portion of the picture and expands it to large size. The individual pixels that make up the screen are expanded to about 2 millimeters on a side. The user is able to turn on and off individual pixels. The picture on the screen bears some resemblance to the original picture, but it is definitely not the same. Nor is it entirely clear how the user got into this mode.

Every user of MacPaint has at one time or another activated this feature. The first time it happens, it is quite confusing. This was the case with many of our subjects. They got into the Fatbits option inavertently and the results were confusing for them. Interestingly enough, all were able to recover and go on to draw completely acceptable houses. The subjects who got into this option were the ones who explored successfully. The less successful explorers didn't get into this kind of confusion. Whether they would have been able to get out of it is an open question. In general, the Macintosh provided a reasonable and challenging exploratory environment.

The next set of comments are general observations on the ways in which subjects explored. The sample that we used was not large enough to break down into specific sets of statistically valid observations. It was intentionally chosen as a diverse sample to see what kinds of exploration we could elicit. In future studies, it would be wise to choose homogeneous groups from the target population of interest. The results of the experiments presented here could offer some guidance in what to look for and how to choose these groups.

Several things were obvious in reviewing the exploration tapes dealing with older subjects. In the first place, these subjects did not tend to explore the options available. They seemed to be far more task oriented and this seemed to inhibit exploration. Their main function seemed to be focusing on getting the job done. Exploration was regarded as something that inhibited performance of the task. They would find a few satisfactory icons and then use these in brute force mode to get the job done. The idea seemed to be "stick with what you know and show that you are working".

This is an attitude that is frequently seen among users of computer equipment. It is a self defeating attitude because usage of computer equipment involves a high degree of learning. Perhaps 25% of the job should be spent in learning or exploration. Such subjects are likely to be trainable in a specific narrow area of computing. They are likely to have trouble broadening out and expanding their horizons to include new technologies.

The different problems surfaced when we were dealing with high status subjects. Such subjects seemed to have difficulty in taking feedback from the machine. Since these tended to be older subjects as well, this did compound some of the observations. Such subjects are used to giving vague commands to secretaries. Secretaries are then responsible for filling in the details of the command and producing the desired result. The subjects themselves are not responsible for the details.

We ran into one subject who needed approval from the experimenter. This subject had to ask before any selection was made whether she was doing it right. She seemed to be quite disturbed with the idea that exploration was all right. This subject was a young girl, and had probably been socialized into the idea that she should be "good". This meant that just being turned loose on an expensive computer with only vague instructions about what to do was not entirely acceptable.

It was interesting to contrast the attitude of this subject with the attitude of a young boy, about one year older. The boy's performance on the machine was not substantially different from the young girl's. The attitudes, however, were miles apart. They young boy took the attitude that he knew what he was doing and how to do it and needed absolutely no help or instruction from the experimenter. Our timid subject was the only such subject we observed in this experimental group, although any teacher of computer science has seen similar subjects in the student population. Such individuals rarely become good at computing and are poor candidates for any kind of work in technological literacy.

Another subject that we observed is what I would classify as a conservative non-user. This person is not a regular user of computing equipment but needs to use it occasionally for employment reasons. Doing so, this individual has developed a conservative strategy. This strategy includes taking small exploratory steps, saving intermediate output, and proceeding from one stable state to another. The subject was willing to indulge in limited backtracking, but only enough to perform the task at hand. This subject's strategies were quite effective and worked well for the individual. It is unlikely that this individual will become a true computer "power user" but the strategy works well for somebody who has limited need of computing power.

The secretarial staff constituted the most effective group of subjects on the Macintosh. Their educational level was below that of the faculty and computer center subjects, but they were able to use the machine extremely effectively. They were used to using word processing equipment, although this equipment was very different from the Macintosh. There is nothing in their ordinary jobs skills that related directly to the tests they were asked to do. The most notable feature of the secretarial subjects was their attention to detail and their willingness to take feedback from the machine. These are tasks that are required in ordinary word processing, and there appeared to be some transfer in the way they used the Macintosh. These are skills that should probably be included in any attempt to teach technological literacy. Whether these skills can be successfully transfered to high status subjects is another question for another research project.

Conclusions and Areas for Future Research

In our subjects, it seemed that it would be a mistake to look for inherent skills or intelligence levels as a factor in performing the assigned tasks. All subjects were intelligent enough to be able to handle it. The difference is in performance level seem to be related to several things, motivation, attitude, and attention. Motivation seemed to be a primary factor in the performance of many subjects. Those who took a nonchalant attitude towards the task did not explore much and as a consequence performed more poorly on the assigned task.

Attitude about the task also was important. Those who took a narrowly focused attitude towards the task did not indulge in much exploration. As a consequence they did not discover the features they needed to fully complete the assigned task. The final factor was the attention to detail. If the subject was not used to paying attention to detail or not used to paying attention to feedback on the screen, the performance in the task was not as good.

The issue is the extent to which this experiment was relevant to the larger areas of technological literacy. This is one that still unanswered. The range of tasks under the heading of technological literacy is so broad that the same objection could be raised to any experiment. Does this task embody the features that one would expect to find in a programming task, a mechanical diagnosis task, or the operation of some complex piece of equipment? My intuitive feeling after watching people perform on this is that it probably does but there is no way to prove this. This experiment was largely exploratory, although it does have implications for both the teaching of technological literacy and for future research in this area.

We feel strongly that exploration is a suitable paradigm for beginning the study of technological literacy. Carroll's taxonomy of exploratory environment characteristics given in Figure 2 is a reasonable place to start. To some degree or another, this paradigm characterises all exploratory environments of interest in technological literacy. All such environments have a certain responsiveness. When you do something you get some feedback. Feedback may be a change on the screen, a change in the performance of an engine, or some other system characteristic. We observed that it is necessary to tell the users how to pay attention to this feedback. For many of our subjects, attention to feedback was a problem.

Related to responsiveness is the idea of benchmarks. A subject needs to know where he is in a given episode or session. A course on technological literacy would also include how to pay attention to these benchmarks.

Acceptable uncertainty and safe conduct are two characteristics of an exploratory environment. The subject needs to know that it is all right to be less than fully confident of your understanding. The courses that subjects have had in technical areas lead them to believe there is one right answer for everything. Mathematics is a common problem in this respect. Students are always feeling like nothing they ever do is right. When they carry this attitude over to an exploratory environment, it very much inhibits their ability to work effectively in that environment.

Acceptable uncertainty and safe conduct emphasize that there is nothing that you can do wrong in this environment that cannot be fixed. A course in technological literacy would want to emphasize these characteristics of an exploratory environment.

The last four elements in the paradigm emphasize the way in which a subject learns in an exploratory environment. These include learning by doing, opportunity, taking charge and control. The subject needs to know that learning is an active and not a passive process. Much of teaching that people are subjected to is of the passive sort. You go, you listen to somebody talk, you take a test and that is it.

This does not work in an exploratory environment and is not an acceptable approach to acquiring technological literacy. Any course that claims to teach somebody technological literacy must emphasize these features of the environment. Even more so, it must emphasize that it is acceptable for subjects to proceed in this way.

One problem we observed was that some subjects did not feel it was acceptable to learn in a working environment. Learning by doing was foreign to them. The only acceptable thing in a work environment was to put your blinders on and charge. Such subjects did not seem to be able to take charge. They did not seem to have the feeling that they were in control of the process.

Our initial goal of trying to find a mental image that contributed to the subject's ability to sclve problems was a failure. There was nothing that we could see in the mental images that the subjects had that had any effect on performance. Far more important than mental image was the subject's attitudes in the experiment.

Our subjects seemed to be setting goals implicitly. Unfortunately, we did not develop a measure for the way that these goals affected their performance in the task. Some of our subjects appeared to have highly task oriented goals and these led away from any acceptance of explorations. The subject's attitude seemed to be far more important than any kind of mental image. A measure of attitude, especially attitude related to work, would probably be far more useful as a way of predicting task performance.

Another feature that should be looked at is the affective component that the subject brings to the experiment. In designing a course for technological literacy, reinforcement of the positive affect may be very useful. Subjects have had many, many years of reinforcement of looking for the right answer in a course. There is no right answer in exploration. There are only effective or ineffective ways of reaching a goal.

Perhaps it would be useful for a course in technological literacy to bolster the confidence of the students. It should try to undo the damage done by many years of conventional teaching. It is probably a mistake in designing courses for people to look for aptitude. It would seem based on these preliminary results that attitude is far more important.

If I were designing a course for technological literacy, these are the points I would consider:

1. Teaching general problem solving skills is likely to be more useful than specific skills. But these general problem solving skills should be taught in the context of a course, and they should be course specific.

2. A course should reward for exploration rather than for the right answer.

3. The course should teach students to pay attention to the feedback that they get from the environment.

4. The course should bolster confidence in the students and teach them that it is all right to make mistakes. It should emphasize the forgiving nature of an exploratory environment.

5. The course should emphasize that the learning process is something that takes place between the student and the machine. It is not something that takes place between the student and the instructor. This is very much a process of learning by doing.

6. The course should bolster the idea of the generalizablity of the material. It should emphasize that the material presented can be used in a variety of areas.

Future research in this area should look at a variety of points. These are: First, it should develop a taxonomy of subjects of various groups of interest. Such groups would include grouping by age, sex, and prior background. Two, within these various interest groups, try to develop certain taxonomies of problem solving styles. These could include willingness to explore, different approaches to exploration. Three, once these have been developed, testing could begin on a variety of experimental treatments for teaching exploration or technological literacy skills.

Appendix A

Instructions to the Subject

Subject Instructions

This is a Macintosh Computer. The computer consists of the screen, to display the computer's response to your commands, and a keyboard and a "mouse", which are instruments used to command the computer to do what you wish. In the present experiment, you will be using only the mouse to communicate to the computer what you want it to do. The mouse is used to position the cursor on the computer screen. To issue your command to the computer simply push the button located on top of the mouse. In turn, the computer will respond accordingly by displaying what you commanded it to.

This experiment should be fun. Look at it as an adventure in exploring the world of the computer. All you are required to do is to draw the picture of the house (show subject the house). You must try to draw the house using the computer exactly the way it appears in the picture. I will not provide any hints as to how the picture can be drawn. You will not be judged, there are no right or wrong answers. The purpose of this experiment is merely to observe how people explore new computer environments. You will have 45 minutes to complete this picture.

Let me explain in more detail how the mouse works. Simply move the mouse around on the table beside your computer screen and you will see the cursor move on the screen. Push the button on the mouse once over the picture showing the function you wish to select and push the button twice to execute the selected function (demonstrate the movement of the cursor with the mouse on the screen).

In case you discover you have made an error and you want to erase a portion of what you have drawn, simply position the cursor with the mouse over the picture of the eraser on the menu and push the button on the mouse once to select the function. Then move the cursor over the mistake in your drawing and move the cursor back and forth as if you were erasing a chalkboard. To erase the entire picture, position the cursor over the picture of the eraser and push the button on the mouse twice.

Are there any questions?

Note - the experimenter demonstrates the use of the mouse to select an icon, draw a line, access a pull-down menu and erase a whole picture.

Appendix B

COMPUTER EXPLORATION PRETEST

Name: ________________________ Date: ____________________

1. How old are you?
2. How many years of schooling have you had?
3. Do you like to play games?
4. What games do you like to play?
   a) Chess
   b) Bridge
   c) Go
   d) Poker
5. Do you like video games?
   a) Space Invaders
   b) PacMan
   c) Asteroids
   d) Adventure type games
   e) Other
6. Are you comfortable with mathematics?
7. What mathematics courses have you taken?
   a) Arithmetic (years?)

b) Algebra

c) Geometry

d) Trigonometry

e) Calculus

f) Logic

g) Other

8. Do you play a musical instrument?

   a) Instrument

   b) Years

9. Do you read music?

10. Do you type?

    a) How fast?

11. Have you ever used a word processor?

12. Do you own a personal computer?

13. Have you ever used a computer?

14. What computer languages have you ever used?

    a) BASIC

    b) PASCAL

    c) FORTRAN

    d) LOGO

    e) Other

15. Have you ever used a Macintosh?

Appendix C

Posttest Interview

Is this computer like anything you have used before?

- Typewriter?
- Pencil and paper?
- Paintbrush?
- Drafting tools?
- Other computers and wordprocessors?

What would you use this computer for?

What could make this computer easier to use?

Appendix D

Subject Notes

Subject Number: 1
Sex: F Age: 43 Years of School: 20

Likes games:	1
Types:	
Video Games:	0
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1

Typing Skills

Typewriter:	1
Speed:	70
Word Processor:	1
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PASCAL, FORTRAN, COBOL, SPSS, SAS
Used a Macintosh:	0
House Score:	15

House Feature	Score
Walls	
Shape	1
Texture	1
Border	
Ridgeline	
Shape	
Texture	
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	

Picture window	
Shape	
Texture	
Border	
Door	
Shape	1
Texture	1
Border	
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	
Texture	
Roof	
Shape	
Texture	1
Border	
Chimney	
Shape	1
Texture	1
Border	
Smoke	
Use of Pull-Down Menu	
Texture	1
Total	15

Notes - The subject felt that the primary goal was to reproduce the house. The subject did not really feel free to explore. Work drove the experiment. The subject used a brute force approach and tried to recreate icon functions manually. This took most of the allotted time for the experiment. The subject began exploration toward the end of the experiment.

Subject Number: 2
Sex: M Age: 33 Years of School: 19

Likes games:	0
Types:	
Video Games:	0
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1

Typing Skills

Typewriter:	1
Speed:	20
Word Processor:	1
Own a PC:	1
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PASCAL, FORTRAN, LISP, APL
Used a Macintosh:	0
House Score:	25

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	1
Texture	1
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	1
Texture	
Border	1
Door knob	
Shape	
Texture	1
Roof	
Shape	1
Texture	1
Border	1
Chimney	
Shape	1
Texture	1
Border	
Smoke	
Use of Pull-Down Menu	1
Texture	1
Total	25

Subject Number: 3
Sex: M Age: 38 Years of School: 18

Likes games:	1
Types:	1
Video Games:	1
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1
Typing Skills	
Typewriter:	1
Speed:	30
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PASCAL, FORTRAN, SPSS, SAS
Used a Macintosh:	0
House Score:	28

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	1
Texture	1
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	
Texture	1
Roof	
Shape	1
Texture	1
Border	1
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	1
Texture	1
Total	28

Subject Number: 4
Sex: F Age: 32 Years of School: 13

Likes games:	1
Types:	Chess, Poker
Video Games:	1
Comfortable with Math:	0
Math Courses:	Arithmetic
Musical Instrument:	0

Typing Skills

Typewriter:	1
Speed:	75
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	
Used a Macintosh:	0
House Score:	22

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	
Texture	1
Roof	
Shape	1
Texture	1
Border	1
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	
Total	22

Notes - Subject initially used only the demonstrated functions and seemed reluctant to try others. The subject discovered the most versatile icon (rubber-band lines) early and used it to draw the whole house.

Subject Number: 5
Sex: M Age: 37 Years of School: 20

Likes games:	1
Types:	Poker
Video Games:	0
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	0
Typing Skills	
Typewriter:	0
Speed:	0
Word Processor:	1
Own a PC:	1
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PASCAL, FORTRAN, C
Used a Macintosh:	0
House Score:	18

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	
Texture	
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	
Border	1
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	
Total	18

Subject Number: 6
Sex: F Age: 29 Years of School: 12

Likes games:	1
Types:	Chess, Poker
Video Games:	1
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry
Musical Instrument:	0

Typing Skills

Typewriter:	1
Speed:	50
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC, COBOL
Used a Macintosh:	0
House Score:	12

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	
Texture	1
Border	
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	
Picture window	
Shape	
Texture	
Border	
Door	
Shape	
Texture	

Border	
Door window	
Shape	
Texture	
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	
Texture	1
Border	1
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	
Total	12

Notes - Subject experimented with the demo functions first and decided to fill in background texture first and then erase into a rectangular shape. Conservative - selected a strategy early and stuck with it.

Subject Number: 7
Sex: M Age: 8 Years of School: 2

Likes games:	1
Types:	
Video Games:	1
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1

Typing Skills

Typewriter:	0
Speed:	
Word Processor:	0
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	
Used a Macintosh:	0
House Score:	13

House Feature	Score
Walls	
Shape	1
Texture	
Border	
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	
Door	
Shape	1
Texture	1

Border	
Door window	
Shape	1
Texture	1
Border	
Door knob	
Shape	1
Texture	
Roof	
Shape	1
Texture	
Border	
Chimney	
Shape	1
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	13

Subject Number: 8
Sex: F Age: 24 Years of School: 13

Likes games:	1
Types:	Chess
Video Games:	1
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra
Musical Instrument:	1

Typing Skills

Typewriter:	1
Speed:	30
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PYXEL VISUALS
Used a Macintosh:	0
House Score:	20

House Feature	Score
Walls	
Shape	1
Texture	
Border	1
Ridgeline	
Shape	1
Texture	1
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	
Texture	
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	1
Border	1
Chimney	
Shape	1
Texture	
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	
Total	20

Subject Number: 9

Sex: M Age: 32 Years of School:	19
Likes games:	1
Types:	Bridge
Video Games:	0
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1
Typing Skills	
Typewriter:	1
Speed:	30
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC, FORTRAN, SPSS, SAS
Used a Macintosh:	0
House Score:	23

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	1
Texture	1
Border	
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	1
Border	
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	
Total	23

Subject Number: 10
Sex: M Age: 45 Years of School: 16

Likes games:	0
Types:	
Video Games:	0
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig Statistics
Musical Instrument:	0

Typing Skills

Typewriter:	1
Speed:	30
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	FORTRAN, COBOL, PL/I, RPG, AUTOCODER
Used a Macintosh:	0
House Score:	7

House Feature	Score
Walls	
Shape	
Texture	1
Border	
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	
Texture	1
Border	
Door	
Shape	
Texture	1

Border	
Door window	
Shape	
Texture	1
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	
Texture	1
Border	
Chimney	
Shape	
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	7

Notes - the subject demonstrated a lack of interest in exploration in the experiment that is consistent with his professional work habits.

Subject Number: 11
Sex: F Age: 26 Years of School: 15

Likes games:	1
Types:	Board games
Video Games:	1
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Statistics
Musical Instrument:	1

Typing Skills

Typewriter:	1
Speed:	30
Word Processor:	1
Own a PC:	1
Used a PC:	1
Used a computer:	1
Languages:	BASIC
Used a Macintosh:	0
House Score:	24

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	1
Texture	1
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	
Door	
Shape	1
Texture	1

Border	
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	1
Texture	1
Roof	
Shape	1
Texture	1
Border	
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	1
Texture	1
Total	24

Subject Number: 12
Sex: M Age: 68 Years of School: 16

Likes games:	0
Types:	
Video Games:	0
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig
Musical Instrument:	0

Typing Skills

Typewriter:	0
Speed:	0
Word Processor:	0
Own a PC:	0
Used a PC:	0
Used a computer:	0
Languages:	
Used a Macintosh:	0
House Score:	12

House Feature	Score
Walls	
Shape	
Texture	1
Border	
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	
Texture	1
Border	
Door	
Shape	
Texture	1

Border	
Door window	
Shape	
Texture	1
Border	
Door knob	
Shape	
Texture	1
Roof	
Shape	
Texture	1
Border	
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	1
Total	12

Subject Number: 13
Sex: M Age: 51 Years of School: 20

Likes games:	1
Types:	Chess, Bridge
Video Games:	0
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1

Typing Skills

Typewriter:	1
Speed:	40
Word Processor:	0
Own a PC:	0
Used a PC:	0
Used a computer:	1
Languages:	FORTRAN
Used a Macintosh:	0
House Score:	18

House Feature	Score
Walls	
Shape	1
Texture	1
Border	
Ridgeline	
Shape	1
Texture	1
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	
Texture	1
Border	
Door knob	
Shape	
Texture	1
Roof	
Shape	1
Texture	1
Border	
Chimney	
Shape	1
Texture	1
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	18

Notes - Subject is reluctant to explore. Tries to find function by thinking

Subject Number: 14
Sex: M Age: 15 Years of School: 10

Likes games:	1
Types:	Chess
Video Games:	1
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra
Musical Instrument:	0

Typing Skills

Typewriter:	1
Speed:	15
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	0
Languages:	BASIC, LOGO
Used a Macintosh:	0
House Score:	27

House Feature	Score
Walls	
Shape	1
Texture	1
Border	1
Ridgeline	
Shape	1
Texture	1
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	1
Texture	1
Roof	
Shape	1
Texture	1
Border	1
Chimney	
Shape	1
Texture	1
Border	1
Smoke	
Use of Pull-Down Menu	
Texture	
Total	27

Subject Number: 15
Sex: F   Age: 38   Years of School: 14

Likes games:	0
Types:	
Video Games:	0
Comfortable with Math:	1
Math Courses:	Arithmetic, Algebra, Geometry
Musical Instrument:	0

Typing Skills

Typewriter:	0
Speed:	70
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC
Used a Macintosh:	0
House Score:	10

House Feature	Score
Walls	
Shape	1
Texture	
Border	
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	
Picture window	
Shape	1
Texture	1
Border	
Door	
Shape	1
Texture	1

Border	
Door window	
Shape	1
Texture	1
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	
Texture	
Border	
Chimney	
Shape	1
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	0

Subject Number: 16
Sex: M Age: 36 Years of School: 14

Likes games:	1
Types:	Poker
Video Games:	0
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra
Musical Instrument:	0

Typing Skills

Typewriter:	1
Speed:	80
Word Processor:	1
Own a PC:	0
Used a PC:	0
Used a computer:	0
Languages:	
Used a Macintosh:	0
House Score:	11

House Feature	Score
Walls	
Shape	1
Texture	
Border	1
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	
Picture window	
Shape	1
Texture	1
Border	
Door	
Shape	1
Texture	1

Border	
Door window	
Shape	
Texture	
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	
Border	1
Chimney	
Shape	
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	11

Subject Number: 17
Sex: M Age: 45 Years of School: 16

Likes games:	0
Types:	
Video Games:	0
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	0
Typing Skills	
Typewriter:	1
Speed:	30
Word Processor:	1
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PASCAL, FORTRAN, PL/I
Used a Macintosh:	0
House Score:	3

House Feature	Score
Walls	
Shape	1
Texture	1
Border	
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	
Texture	
Border	
Crosspieces	
Picture window	
Shape	
Texture	
Border	
Door	
Shape	
Texture	

Border	
Door window	
Shape	
Texture	
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	
Border	
Chimney	
Shape	
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	3

Subject Number: 18
Sex: M Age: 52 Years of School: 17

Likes games:	0
Types:	
Video Games:	0
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic
Musical Instrument:	0
Typing Skills	
Typewriter:	1
Speed:	30
Word Processor:	0
Own a PC:	0
Used a PC:	1
Used a computer:	1
Languages:	BASIC, PASCAL, FORTRAN, ASSEMBLER
Used a Macintosh:	0
House Score:	15

House Feature	Score
Walls	
Shape	1
Texture	1
Border	
Ridgeline	
Shape	1
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	
Door	
Shape	1
Texture	1

Border	
Door window	
Shape	1
Texture	1
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	
Border	
Chimney	
Shape	1
Texture	1
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	15

Subject Number: 19

Sex: M Age: 36 Years of School:	20
Likes games:	1
Types:	Chess, Bridge, Go
Video Games:	1
Comfortable with Math:	0
Math Courses:	Arithmetic, Algebra, Geometry, Trig, Calculus, Logic, Statistics
Musical Instrument:	1
Typing Skills	
Typewriter:	1
Speed:	35
Word Processor:	1
Own a PC:	1
Used a PC:	1
Used a computer:	1
Languages:	BASIC, FORTRAN
Used a Macintosh:	0
House Score:	19

House Feature	Score
Walls	
Shape	1
Texture	
Border	1
Ridgeline	
Shape	1
Texture	
Border	1
Small window	
Shape	1
Texture	1
Border	1
Crosspieces	1
Picture window	
Shape	1
Texture	1
Border	1
Door	
Shape	1
Texture	1

Border	1
Door window	
Shape	1
Texture	1
Border	1
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	
Border	1
Chimney	
Shape	
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	19

Subject Number: 20 Sex: F Age: 7 Years of School: 1

Likes games:	1
Types:	
Video Games:	1
Comfortable with Math:	1
Math Courses:	Arithmetic
Musical Instrument:	1 Typing Skills
Typewriter:	0
Speed:	0
Word Processor:	0
Own a PC:	1
Used a PC:	1
Used a computer:	1
Languages:	
Used a Macintosh:	0
House Score:	8

House Feature	Score
Walls	
Shape	
Texture	
Border	
Ridgeline	
Shape	
Texture	
Border	
Small window	
Shape	1
Texture	1
Border	
Crosspieces	1
Picture window	
Shape	
Texture	1
Border	
Door	
Shape	
Texture	1
Border	

Door window	
Shape	
Texture	1
Border	
Door knob	
Shape	
Texture	
Roof	
Shape	1
Texture	
Border	
Chimney	
Shape	1
Texture	
Border	
Smoke	
Use of Pull-Down Menu	
Texture	
Total	8

BIBLIOGRAPHY

Anzai, Yuichiro and Simon, Herbert A., "The Theory of Learning by Doing", *Psychological Review*, Vol. 86, No. 2, pp. 124-140.

Baird, Leonard L., *Review of Problem-Solving Skills*, Educational Testing Service Report No. ETS-RR-83-16, Princeton, NJ, March 1983.

Bayman, Pierre and Mayer, Richard E., "A Diagnosis of Beginning Programmers' Misconceptions of BASIC Programming Statements", *Communications of the ACM*, Vol. 26, No. 9, September 1983, pp. 677-679.

Bell, M. S., "Applied Problem Solving as a School Emphasis: An Assessment and Some Recommendations", pp. 5-24 in Lesh, R., Mierkiewics, D., and Kantowski, M. (eds.), *Applied Mathematical Problem Solving*, ERIC Clearinghouse for Science, Mathematics and Environmental Education, Columbus, Ohio, 1979.

Bhaskar, R. and Simon, H. A., "Problem solving in semantically rich domains: An example from engineering thermodynamics", *Cognitive Science*, Vol. 1, 1977, pp. 193-215.

Bourne, Lyle, "Knowing and using concepts", *Psychological Review*, Vol. 77, No. 6, November 1970, pp. 546-556.

Brooks, R. E., "Toward a Theory of the Cognitive Processes in Computer Programming", *International Journal of Man-Machine Studies*, 1977, pp. 737-751.

Brooks, R. E., "Studying programmer behavior experimentally: The problems of proper methodology", *Communications of the ACM*, Vol. 23, No. 4, April 1980, pp. 207-213.

Brown, Ann L.," Knowing When, Where, and How to Remember: A Problem of Metacognition", Technical Report No. 47, Bolt, Beranek and Newman, Inc., Cambridge, Mass. and Center for the Study of Reading, University of Illinois, Urbana, Illinois, June 1977.

Brown, P. J., "Error Messages: The Neglected Area of the Man/Machine Interface", *Communications of the Association for Computing Machinery*, Vol. 26, No. 4, April 1983, pp. 246-249.

Carroll, John M. and Mack, Robert L., "Learning to Use a Word Processor: By Doing, By Thinking and By Knowing", Research Report RC 9481 (41921) 7/26/82, Computer Science Department, IBM Watson Research Center, Yorktown Heights, NY 10598.

Carroll, John M., "The Adventure of Getting to Know a Computer", *Computer*, November 1982, pp. 49-58.

Carroll, John M. and Mack, Robert L. "Actively Learning to Use a Word Processor", Chapter 11 in W. E. Cooper (ed), *Cognitive Aspects of Skilled Typewriting*, Springer Verlag, New York, 1983.

Carroll, John M. and Carrithers, Caroline "Blocking User Error States in a Training System", Research Report RC 10100 (44894) 8/2/83, Computer Science Department, IBM Watson Research Center, Yorktown Heights, NY 10598.

Carroll, John M., Mack, Robert L., Lewis, Clayton, Grischkowsky, Nancy L., and Robertson, Scott R., "Learning to Use a Word Processor by Guided Exploration", Research Report RC 10428 (46607) 3/20/84, Computer Science Department, IBM Watson Research Center, Yorktown Heights, NY 10598.

Carroll, John M. and Mazur, Sandra A. "Learning to Use an Office System With an On-Line Tutorial", Research Report RC 10644 (47740) 7/26/84, Computer Science Department, IBM Watson Research Center, Yorktown Heights, NY 10598.

Chi, Michelene T. H., Feltovich, Paul J., and Glaser, Robert, "Categorization and Representation of Physics Problems by Experts and Novices", *Cognitive Science*, Vol. 5, 1981, pp. 121-152.

Christensen-Szalanski, J. J., "A further examination of the selection of problem-solving strategies: The effects of deadlines and analytic aptitudes", *Organizational Behavior and Human Performance*, Vol. 25, 1980, pp. 107-122.

Cyrevt, R. M., "Problem Solving and Educational Policy", pp. 3-8 in Tuma, D. T., and Reif, F. (Eds.), *Problem Solving and Education: Issues for Teaching and Research*, Earlbaum, Hillsdale, New Jersey, 1980.

Dadourian, H. M., *How to Study, How to Solve*, Addison Wesley, Cambridge, Massachusetts, 1949.

Darlington, Jared, Dzida, Wolfgang and Herda, Siegfried, "The Role of Excursions in Interactive Systems", *International Journal of Man-Machine Studies*, Vol. 18, pp. 101-112, 1983.

Davis, G. A., "Current Status of Research and Theory in Human Problem Solving", *Psychological Bulletin*, Vol. 66, 1966, pp. 36-54

Davis, G. A., *Psychology of Problem Solving*, Basic Books, New York, 1973.

Dery, David, "Decision-making, Problem Solving and Organizational Learning", *Omega*, Vol 11, No. 4, 1983, pp. 321-328.

Elstein, A. S., Shulman, L. S., Sprakfa, S. A., *Medical Problem Solving: An Analysis of Clinical Reasoning*, Harvard University Press, Cambridge, Massachusetts, 1978.

Ericsson, K. A. and Simon, H. A., "Sources of evidence on cognition: A historical overview", in Merluzzi, T. V., Glass, C. R. and Genest, M. (eds.), *Cognitive Assessment*, Guilford Press, New York, 1981.

Ericsson, K. A., and Simon, H. A., "Verbal reports as data", *Psychological Review*, Vol. 87, No. 3, May 1980, pp. 215-251.

Fredriksen, N. and Ward, W. C., "Measures for the Study of Creativity in Scientific Problem-Solving", *Applied Psychological Measurement*, Vol. 2, 1978, pp. 1-24.

Getzels, J. W. and Csikszentmihalyi, M. "From Problem Solving to Problem Finding", pp. 90-116 in Taylor, I. A. and Getzels, J. W. (eds.), *Perspectives in Creativity*, Aldine, Chicago, 1975.

Good, Michael D., Whiteside, John A., Wixon, Dennis R., and Jones, Sandra J., "Building a User-Derived Interface", *Communications of the Association for Computing Machinery*, Vol. 27, No. 10, October 1984, pp. 1032-1043.

Greeno, J. G., "Trends in the Theory of Knowledge for Problem Solving", pp. 9-24 in Tuma, D. T., and Reif, F. (eds.), *Problem Solving and Education: Issues for Teaching and Research*, Earlbaum, Hillsdale, New Jersey, 1980.

Hayes, John R., *The Complete Problem Solver*, The Franklin Institute Press, Philadelphia, Pa., 1981.

Hill, C. C., *Problem Solving: Learning and Teaching: An Annotated Bibliography*, Nichols Publishing, New York, 1979.

Jacob, Robert J. K., "Using Formal Specifications in the Design of a Human-Computer Interface", *Communications of the Association for Computing Machinery*, Vol. 26, No. 4, April 1983, pp. 259-264.

Johnson, P. E., Baretto, A., Hassebrock, F., Moller, J., Prietula, M., Feltovich, P. J. and Swanson, D. B., "Expertise and Error in Diagnostic Reasoning", *Cognitive Science*, Vol. 5, No. 3, 1981, pp. 439-457.

Karat, John, "A model of problem solving with incomplete constraint knowledge", *Cognitive Psychology*, Vol 14, No. 4, October 1982, pp. 538-559.

Kosslyn, Stephen M., Pinker, Steven, Smith, George E. and Schwartz, Steven, "On the demystification of mental imagery", *The Behavioral and Brain Sciences*, Vol. 2, 1979, pp. 535-581.

Kosslyn, Stephen Michael, "Research on mental imagery: Some goals and directions", *Cognition*, Vol. 10, 1981. pp. 173-179.

Larkin, J. H., "Teaching Problem Solving in Physics: The Psychological Laboratory and the Practical Classroom", pp. 111-126 in Tuma, D. T., and Reif, F. (eds.), *Problem Solving and Education: Issues for Teaching and Research*, Earlbaum, Hillsdale, New Jersey, 1980.

Larkin, J. H., McDermott, J. Simon, D. P. and Simon, H. A., "Expert and Novice Performance in Solving Physics Problems", *Science*, 208:4403, 1980, pp. 1325-1342.

Lesh, R., Mierkiewics, D., and Kantowski, M. (eds.), *Applied Mathematical Problem Solving*, ERIC Clearinghouse for Science, Mathematics and Environmental Education, Columbus, Ohio, 1979.

Lewis, Clayton and Mack, Robert L. "The Role of Abduction in Learning to Use a Computer System", Research Report RC 9433 (41620) 6/15/82, Computer Science Department, IBM Watson Research Center, Yorktown Heights, NY 10598.

Lukas, George et. al., *LOGO Teaching Sequences on Strategy in Problem-Solving and Story Problems in Algebra. Teacher's Text and Problems*, Bolt, Beranek and Newman, Inc., Cambridge, Mass, 30 June 1971.

Mack, Robert L., Lewis, Clayton H. and Carroll, John M., "Learning to Use Word Processors: Problems and Prospects", *ACM Transactions on Office Information Systems*, Vol. 1, No. 3, pp. 254-271, July 1983.

Malone, Thomas W., "Toward a Theory of Intrinsically Motivating Instruction", *Cognitive Science*, Vol. 4, 1981, pp. 333-369.

Malone, Thomas W., "What Makes Computer Games Fun?", *Byte*, Vol. 6, No. 12, December 1981, pp. 258-277.

Malone, Thomas W., "What Makes Things Fun to Learn? A Study of an Intrinsically Motivating Computer Game", Paper presented at the American Educational Research Association Annual Meeting, Los Angeles, California, April 13-17, 1981.

Malone, Thomas W., "Heuristics for Designing Enjoyable User Interfaces: Lessons from Computer Games", in *Proceedings of Human Factors in Computer Systems*, Association For Computing Machinery, Gaithersburg, Maryland, March 15-17, 1982.

Maltzman, I., "On the Training of Originality", *Psychological Review*, Vol. 67, 1960, pp. 229-242.

Mayer, R. E., "Different problem solving competencies established in learning computer programming with and without meaningful models", *Journal of Educational Psychology*, Vol. 67, 1975, pp. 725-734.

Mayer, Richard E., "Some Conditions of Meaningful Learning for Computer Programming: Advance Organizers and Subject Control of Frame Order", *Journal of Educational Psychology*, Vol. 68, No. 2, 1976, pp.143-150.

Mayer, R. E., *Thinking and Problem Solving: An Introduction to Human Cognition and Learning*, Scott-Foresman, Glenview, Illinois, 1977.

Mayer, R. E., "A Psychology of Learning BASIC", *Communications of the ACM*, Vol. 22, No. 11, November 1979, pp. 589-593.

Mayer, R. E., "The Psychology of How Novices Learn Computer Programming", *Computing Surveys*, Vol. 13, No. 1, March 1981, pp. 121-141.

Mayer, R. E. and Bayman P., "Psychology of Calculator Languages: A Framework for Describing Differences in Users' Knowledge", *Communications of the ACM*, Vol. 24, No. 8 August 1981, pp. 511-520.

Moran, T. P., "An Applied Psychology of the User", *Computing Surveys*, Vol. 13, No. 1, March 1981, pp. 1-11.

Morland, D. Verne, "Human Factors Guidelines for Terminal Interface Design", *Communications of the Association for Computing Machinery*, Vol. 26, No. 7, July 1983, pp. 484-494.

Newell, Alan and Simon, Herbert A., *Human Problem Solving*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.

Norman, Donald A., "Design Rules Based on Analyses of Human Error", *Communications of the Association for Computing Machinery*, Vol. 26, No. 4, April 1983, pp. 254-258.

Pitt, R. B., "Development of a general problem-solving schema in adolescence and early adulthood", *Journal of Experimental Psychology*, Vol. 112, No. 4, 1983, pp. 547-584.

Polya, Georg, *How to Solve It*, Doubleday & Company, Inc., Garden City, New York, 1957.

Polya, Georg, *Mathematical Discovery: On Understanding, Learning and Teaching Problem Solving*, Vol. I and II, John Wiley and Sons, Inc., New York, 1962.

Pounds, William F., "The Process of Problem Finding", *Industrial Management Review*, Fall 1969, pp. 1-19.

Purdue University, Purdue Elementary Problem Solving Inventory, Educational Research Center, Purdue University, Lafayette, Indiana, December 1973.

Raaheim, Kjell and Kaufmann, Geir, "Is there a General Problem Solving Ability?", *Journal of General Psychology*, Vol. 90, No. 2, April 1974, pp. 231-236.

Reif, F. and Heller, Joan I., "Knowledge structure and problem solving in physics", *Educational Psychologist*, Vol. 17, No. 2, Summer 1982, pp.. 102-127.

Reitman, Walter R., *Cognition and Thought*, John Wiley and Sons, New York, 1965.

Rouse, W. B., "A model of human decision making in fault diagnosis task", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-8, No. 5, May 1978, pp. 357-361.

Rouse, W. B., "A model of human decision making in fault diagnosis tasks that include feedback and redundancy", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9, No. 4, April 1979, pp. 237-241.

Rouse, W. B. and Rouse, S. H., "Measures of complexity of fault diagnosis", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9, No. 11, November 1979, pp. 720-727.

Rouse, W. B., Rouse, S. H. and Pelligrino, S. J., "A rule-based model of human problem solving in fault diagnosis tasks", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-10, No. 7, July 1980, pp. 366-376.

Rouse, S. H. and Rouse, W. B., "Computer-based manuals for procedural information", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-10, No. 8, August 1980, pp. 506-510.

Rubenstein, M. F., and Pfeiffer, K., *Concepts in Problem Solving*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1980.

Rubenstein, M. F., "A Decade of Experience in Teaching an Interdisciplinary Problem-Solving Course", pp. 25-38 in Tuma, D. T., and Reif, F. (eds.), *Problem Solving and Education: Issues for Teaching and Research*, Earlbaum, Hillsdale, New Jersey, 1980.

Scriven, M., "Prescriptive and Descriptive Approaches to Problem Solving", pp. 127-140 in Tuma, D. T., and Reif, F. (eds.), *Problem Solving and Education: Issues for Teaching and Research*, Earlbaum, Hillsdale, New Jersey, 1980.

Sheil, B. A., "The Psychological Study of Programming", *Computing Surveys*, Vol. 13, No. 1, March 1981, pp. 101-120.

Shneiderman, Ben, "Designing Computer System Messages", *Communications of the ACM*, Vol. 25, No. 9, September 1982, pp. 610-611.

Shrager, Jeff and Klahr, David, "Learning in an Instructionless Environment: Observation and Analysis", in *Proceedings of the CHI'83 Conference, Human Factors in Computing Systems*, December 12-15, 1981, Boston, Mass., Association for Computing Machinery.

Simon, H. A., "Identifying basic abilities underlying intelligent performance of complex tasks", in Anderson, R. C., Spiro, R. and Montagne, W. (eds), *Schooling and the Acquisition of Knowledge*, Lawrence Earlbaum Associates, 1977.

Speedie, Stuart M., Houtz, John, Ringenbach, Susan, and Feldhusen, John F., "Abilities Measured by the Purdue Elementary Problem-Solving Inventory", *Psychological Reports*, Vol. 33, 1973, pp. 959-963.

Sticht, Tom, "Literacy, Cognitive Robotics and General Technology Training for Marginally Literate Adults", presented at the international conference on The Future of Literacy in a Changing World, University of Pennsylvania, May 9-12, 1985.

Stohr, Edward A., and White, Norman H., "User Interfaces for Decision Support Systems: An Overview", *International Journal of Policy Analysis and Information Systems*, Vol. 6, No. 4, pp 393-423.

Suydam, M. N., "Untangling Clues From Research On Problem Solving", pp. 34-50 in Krulisk, S. and Reys, R. E., (eds.), *Problem Solving in School Mathematics*, National Council of Teachers of Mathematics, Reston, Virginia, 1980.

Sweller, J., "Transfer effects in a problem solving context", *Quarterly Journal of Experimental Psychology*, Vol. 32, No. 2, May 1980, pp. 233-239.

Taylor, I. A. and Getzels, J. W. (eds.), *Perspectives in Creativity*, Aldine, Chicago, 1975.

Tuma, D. T., and Reif, F. (eds.), *Problem Solving and Education: Issues for Teaching and Research*, Earlbaum, Hillsdale, New Jersey, 1980.

Vitalari, Nicholas P. and Dickson, Gary W., "Problem Solving for Effective Systems Analysis: An Exploration", *Communications of the ACM*, Vol. 26, No. 11, November 1983, pp. 948-956.

Wicklegren, W. A., *How to Solve Problems: Elements of a Theory of Problems and Problem Solving*, Freeman, Inc., San Francisco, 1974.

Distribution List

Recipient	No.Copies
Director of Research Code 012 Naval Postgraduate School Monterey, CA 93943	1
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Library Code 1424 Naval Postgraduate School Monterey, CA 93943	2
Professor Norman R. Lyons Code 54LB Naval Postgraduate School Monterey, CA 93943	10
Professor Carl R. Jones Code 54JS Naval Postgraduate School Monterey, CA 93943	1
Professor Norman F. Schneidewind Code 54SS Naval Postgraduate School Monterey, CA 93943	1
Professor Tung Bui Code 54BD Naval Postgraduate School Monterey, CA 93943	1
Professor Dan Dolk Code 54DK Naval Postgraduate School Monterey, CA 93943	1
Professor Taracad Sivasankaran Code 54SE Naval Postgraduate School	1